Estructuras de Datos y Algoritmos

Grados en Ingeniería Informática

Examen 2º PARCIAL JUNIO, 31 de mayo de 2018

1. (2 puntos) Extiende la implementación basada en nodos doblemente enlazados del TAD `Lista` con la siguiente operación:

```
void repartir()
```

Dicha operación debe realizar una transformación de la lista de tal forma que: (i) delante del primer elemento de la lista original se coloquen todos los elementos que sean estrictamente menores que él (a este grupo de elementos que quedan delante del primero de la lista original lo llamaremos *primer tramo*); (ii) después del primer elemento de la lista original permanezcan todos los elementos que son iguales o mayores que él (a este otro grupo de elementos que quedan detrás del primero de la lista original lo llamaremos *segundo tramo*). Así mismo, los elementos de cada uno de los tramos tienen que preservar las posiciones relativas que tenían en la lista original (esto es, si dos elementos, X e Y, están en un mismo tramo y el elemento X aparecía antes que Y en la lista original, X seguirá apareciendo antes que Y en el tramo). Si la lista está vacía, la operación no tendrá ningún efecto.

Por ejemplo, si la lista contiene los siguientes elementos (de principio a fin):

```
8 10 5 19 3 8 4 7 2
```

tras ejecutar dicha operación los elementos de la lista, de principio a fin, serán:

```
5 3 4 7 2 8 10 19 8
```

En la implementación no deben crearse ni destruirse nodos, ni tampoco realizarse asignaciones entre los contenidos de los nodos. Aparte de implementar la operación, debes indicar la complejidad de la misma.

2. (2 puntos) En el desarrollo de un juego de aventuras conversacionales se está representando el mapa del juego mediante un árbol binario de personajes. En cada nodo del árbol hay un personaje (*monstruo*, *caballero* o *dama*) representado a través de un carácter (`'M'`,`'C'` y `'D'`, respectivamente)

Se dice que un nodo está *a salvo* cuando: (i) en él hay una dama; (ii) el número de monstruos que hay en el camino que va desde la raiz a dicho nodo es menor o igual que el número de caballeros que hay en los nodos descendientes de dicho nodo.

Debes implementar una función:

```
int num_a_salvo(const Arbin<char>& mapa);
```

que determine el número de nodos en `mapa` que están *a salvo*. Indica además su complejidad.

3. (6 puntos) Se desea desarrollar una aplicación que permita gestionar un parque natural. En el parque natural hay ecosistemas, cada uno identificado a través de un nombre único. Cada ecosistema contiene un cierto número de ejemplares de cada una de las distintas especies que en él habitan. En un ecosistema no pueden habitar dos especies con igual identificador, aunque una misma especie (igual identificador) sí puede habitar en dos o más ecosistemas distintos. Se pueden añadir nuevos ecosistemas en el parque; en un ecosistema se puede añadir una nueva especie con un cierto número de ejemplares o incrementar el número de ejemplares de una especie ya existente en el ecosistema; se puede consultar las últimas especies nuevas que han pasado a formar parte de un ecosistema; se puede consultar el número de ejemplares que hay de una determinada especie en un ecosistema; se puede obtener un listado ordenado alfabéticamente de todas las especies del parque; por último, se puede consultar el número de ejemplares que hay de una determinada especie en el parque.

Se pide implementar un TAD `ParqueNatural` que proporcione las siguientes operaciones:

- `crea()`: Crea un parque natural vacío.
- `an_ecosistema(ecosistema)`: Añade un nuevo ecosistema con identificador `ecosistema` al parque. Si el ecosistema ya existe, levanta una excepción `EEcosistemaDuplicado`.
- `an_ejemplares(ecosistema,especie,n)`: Añade `n` ejemplares de la especie con identificador `especie` al ecosistema con identificador `ecosistema`. Si `ecosistema` no existe, levanta una excepción `EEcosistemaNoExiste`. Si la especie ya habita en el ecosistema se incrementará su número de ejemplares en `n`; si la especie no habita en el ecosistema, se registrará en el ecosistema esa nueva especie con ese número de ejemplares.
- `lista_especies_ecosistema(ecosistema,n)->lista`: Devuelve una lista con los identificadores de las `n` últimas nuevas especies añadidas al ecosistema de identificador `ecosistema`, ordenadas por orden inverso de inserción (es decir, primero la última nueva añadida, segundo la penúltima nueva añadida, y así sucesivamente). Si `ecosistema` no existe, levanta una excepción `EEcosistemaNoExiste`.
- `numero_ejemplares_en_ecosistema(ecosistema,especie)->numero`: Devuelve la cantidad de ejemplares de la especie `especie` que habitan el ecosistema `ecosistema`. Si `ecosistema` no existe, levanta una excepción `EEcosistemaNoExiste`. Si la especie no habita en el ecosistema, la operación devolverá `0` ejemplares.
- `lista_especies_parque()->lista`: Devuelve una lista de todas las especies del parque, ordenada alfabéticamente.
- `numero_ejemplares_en_parque(especie)->numero`: Devuelve el número de ejemplares que hay de la especie `especie` en el parque. Si la especie no habita en el parque, la operación devolverá `0` ejemplares.

Debes elegir la representación del TAD teniendo en cuenta que debes conseguir que la implementación de las operaciones sea lo más eficiente posible. Debes, así mismo, indicar justificadamente la complejidad de cada una de estas operaciones.